动态规划

moongazer

2024 年 02 月 05 日

数据结构的加入

- 当转移有某些好的性质时，我们通常会用各类数据结构去维护来优化，我们通常的做法是，把 dp 值的更新拆分成更新和查询两个部分。
- 一种是边修改边查询，一种是只有初始化和查询
- 前缀和：$f_i = \sum_{l_i \le j \le r_i} f_j$
- 单调队列：AGC007D，NOIP 2017 普及组 跳房子
- 线段树：$f_i = \max_{l_i \le j \le r_i} f_j$

斜率优化——如何观察方程

- HNOI 2008 玩具装箱
- 线性规划的视角和决策单调的视角
- APIO 2010 特别行动队

CDQ 分治

- NOI 2007 货币兑换

矩阵

- Fibonacci number
- 洛谷 P5343
- CF 1286D LCC
- 洛谷 P4223

下面来看点题目吧

洛谷 P2647 最大收益

有 n 个物品，有 W_i 和 R_i 两个属性，每个物品只能选一次，会获得 W_i 的收益，可以选择任意多个物品，但是选择一个后其他的物品的受益会减去 R_i，求能获得的最大收益。
$n \le 3000, W_i, R_i \le 2 \times 10^5$

例题选讲

洛谷 P2647 最大收益

对于固定的若干件物品，最优的方案一定是先选 R_i 较小的，同时发现从前往后选物品需要前面的 R_i 之和但是如果从后往前选只需要知道这是从后往前第 j 个物品，然后把答案减去 $R_i \times (j-1)$。

洛谷 P2647 最大收益

对于固定的若干件物品，最优的方案一定是先选 R_i 较小的，同时发现从前往后选物品需要前面的 R_i 之和但是如果从后往前选只需要知道这是从后往前第 j 个物品，然后把答案减去 $R_i \times (j-1)$。
于是先将物品按照 R_i 从大到小排序，设 $f_{i,j}$ 为前 i 个物品，选了其中 j 个能获得的最大收益，每次新增物品是放到第一个。

一棵 n 个点的树，边有边权。有 m 个关键点 $a_1, a_2, \cdots, a_m$（a_i 是可重的），你需要选择若干个快递中转站 $p_1, p_2, \cdots, p_k$，并为每一个关键点 a_i 找出离它最近的快递中转站 p_{b_i}，试最小化 $k \times C + \sum_{i=1}^{m} dis(a_i, p_{b_i})$，其中 k 是中转站数量，C 是给定的常数，$dis(i, j)$ 表示树上 i 到 j 的路径的边权之和。

考虑 DP，设 $f_{i,j}$ 表示当前对于第 i 个点的子树，还有 j 个关键点没有匹配的最小代价（这里的代价需要计算上没匹配的点连到 i 的代价和），设 $g_{i,j}$ 表示对于第 i 个点的子树，内部的关键点已全部匹配，并且在 j 号你已经选择了（也就是说外面连进来的点都会连向 j）。

例题选讲

寄

考虑 DP，设 $f_{i,j}$ 表示当前对于第 i 个点的子树，还有 j 个关键点没有匹配的最小代价（这里的代价需要计算上没匹配的点连到 i 的代价和），设 $g_{i,j}$ 表示对于第 i 个点的子树，内部的关键点已全部匹配，并且在 j 号你已经选择了（也就是说外面连进来的点都会连向 j）。
考虑转移，对于 $f_{i,j}$ 的转移就是树上背包，对于 $g_{i,j}$ 的转移，我们考虑加入一棵子树后，对于在这棵子树加入前就有的 j，我们可以枚举这棵子树有多少未匹配点的时候是最优的，对于这棵子树里的 j，我们枚举前面加入的未匹配点数，就可以转移了。

例题选讲

寄

考虑 DP，设 $f_{i,j}$ 表示当前对于第 i 个点的子树，还有 j 个关键点没有匹配的最小代价（这里的代价需要计算上没匹配的点连到 i 的代价和），设 $g_{i,j}$ 表示对于第 i 个点的子树，内部的关键点已全部匹配，并且在 j 号你已经选择了（也就是说外面连进来的点都会连向 j）。
考虑转移，对于 $f_{i,j}$ 的转移就是树上背包，对于 $g_{i,j}$ 的转移，我们考虑加入一棵子树后，对于在这棵子树加入前就有的 j，我们可以枚举这棵子树有多少未匹配点的时候是最优的，对于这棵子树里的 j，我们枚举前面加入的未匹配点数，就可以转移了。
考虑复杂度，对于每一对关键点与关键点，关键点与树的结点，它会在 lca 处被枚举到，因此复杂度为 $O(n^2)$。

SHOI 2014 概率充电器

给一棵 n 个点的树，第 i 条边导电的概率为 p_i，第 i 个点直接通电的概率为 q_i，求通电的点个数的期望。

$n \leq 500000$

SHOI 2014 概率充电器

期望个数即为每个点有电概率之和。

例题选讲

SHOI 2014 概率充电器

期望个数即为每个点有电概率之和。
设 u 到 v 的边导电的概率为 $e[u, v]$。
以一个点为根，跑树形 DP 可以求出根通电的概率，设 f_u 为减去 u 到 fa 的边后 u 不通电的概率，
$f_u = (1 - q_u) \times \Pi_v(1 - e[u, v] + e[u, v]f_v)$。

期望个数即为每个点有电概率之和。
设 u 到 v 的边导电的概率为 $e[u, v]$。
以一个点为根，跑树形 DP 可以求出根通电的概率，设 f_u 为减去 u 到 fa 的边后 u 不通电的概率，
$f_u = (1 - q_u) \times \Pi_v(1 - e[u, v] + e[u, v]f_v)$。
这样做要跑 n 次树形 DP，考虑换根。

例题选讲

SHOI 2014 概率充电器

期望个数即为每个点有电概率之和。
设 u 到 v 的边导电的概率为 $e[u,v]$。
以一个点为根，跑树形 DP 可以求出根通电的概率，设 f_u 为减去 u 到 fa 的边后 u 不通电的概率，
$f_u = (1-q_u) \times \Pi_v(1-e[u,v]+e[u,v]f_v)$。
这样做要跑 n 次树形 DP，考虑换根。
设 g_u 为 u 不通电的概率，注意到 g_u 是由 f_u 和父亲的贡献组成的，$\frac{g_{fa}}{1-e[fa,u]+e[fa,u]f_u}$ 即为 g_{fa} 去掉 u 子树不通电的概率

例题选讲

SHOI 2014 概率充电器

期望个数即为每个点有电概率之和。
设 u 到 v 的边导电的概率为 $e[u,v]$。
以一个点为根，跑树形 DP 可以求出根通电的概率，设 f_u 为减去 u 到 fa 的边后 u 不通电的概率，
$f_u = (1-q_u) \times \Pi_v (1 - e[u,v] + e[u,v] f_v)$。
这样做要跑 n 次树形 DP，考虑换根。
设 g_u 为 u 不通电的概率，注意到 g_u 是由 f_u 和父亲的贡献组成的，$\frac{g_{fa}}{1-e[fa,u]+e[fa,u]f_u}$ 即为 g_{fa} 去掉 u 子树不通电的概率
于是 $g_u = f_u \times (1 - e[fa,u] + e[fa,u]\frac{g_{fa}}{1-e[fa,u]+e[fa,u]f_u})$

期望个数即为每个点有电概率之和。
设 u 到 v 的边导电的概率为 $e[u,v]$。
以一个点为根，跑树形 DP 可以求出根通电的概率，设 f_u 为减去 u 到 fa 的边后 u 不通电的概率，
$f_u = (1-q_u) \times \Pi_v(1-e[u,v]+e[u,v]f_v)$。
这样做要跑 n 次树形 DP，考虑换根。
设 g_u 为 u 不通电的概率，注意到 g_u 是由 f_u 和父亲的贡献组成的，$\frac{g_{fa}}{1-e[fa,u]+e[fa,u]f_u}$ 即为 g_{fa} 去掉 u 子树不通电的概率
于是 $g_u = f_u \times (1-e[fa,u]+e[fa,u]\frac{g_{fa}}{1-e[fa,u]+e[fa,u]f_u})$
时间复杂度 $\mathrm{O}(n)$。

例题选讲

CF1415F Cakes for Clones

一条数轴上有一个人，他的移动速度为 1，每个时刻他可以让自己的坐标从 x 变为 $x \pm 1$、或者保持 x 不变。
当他位于整点上时，他可以放置一个无法移动的分身，同时摧毁已经存在的分身（如果存在），放置分身不消耗时间。初始时他位于 0，现在有 n 个任务，每个任务 (x_i, t_i) 要求他在 t_i 时刻要么本人要么分身在 x_i 坐标处。
问是否存在合法方案。

例题选讲

CF1415F Cakes for Clones

首先状态只能是：在 i 位置等蛋糕的同时分身在后面的 j 或者在 i 位置让分身在等 $i-1$ 的蛋糕落下。

例题选讲

CF1415F Cakes for Clones

首先状态只能是：在 i 位置等蛋糕的同时分身在后面的 j 或者在 i 位置让分身在等 $i-1$ 的蛋糕落下。
设 $f_{i,j}$ 为目前在 i 位置分身在 j 位置是否可行，g_i 为位于第 i 个点，前 $i-1$ 个点的蛋糕已经接到了的最小时间。

例题选讲

CF1415F Cakes for Clones

首先状态只能是：在 i 位置等蛋糕的同时分身在后面的 j 或者在 i 位置让分身在等 $i-1$ 的蛋糕落下。
设 $f_{i,j}$ 为目前在 i 位置分身在 j 位置是否可行，g_i 为位于第 i 个点，前 $i-1$ 个点的蛋糕已经接到了的最小时间。
转移就是讨论怎么走，具体可见代码。

给定序列 a，求有多少排列 p 满足：
存在序列 b 满足以下条件：

- $\sum b = m$
- $b_{p_i} \le b_{p_{i+1}}$
- $a_{p_1} + b_{p_1} > a_j$ 或 $a_{p_1} + b_{p_1} = a_j \land p_1 < j$
- $a_{p_i} + b_{p_i} < a_{p_{i+1}} + b_{p_{i+1}}$ 或
  $a_{p_i} + b_{p_i} = a_{p_{i+1}} + b_{p_{i+1}} \land p_i > p_{i+1}$

首先 $=m$ 没有意义，加到最后一个即可。

首先 $=m$ 没有意义，加到最后一个即可。
如果枚举排列 p，b 的构造是显然的贪心。

首先 $=m$ 没有意义，加到最后一个即可。
如果枚举排列 p，b 的构造是显然的贪心。
考虑直接状压，需要记上一个是谁，$\sum b$，上一个 b，复杂度全错。

首先 $=m$ 没有意义，加到最后一个即可。
如果枚举排列 p，b 的构造是显然的贪心。
考虑直接状压，需要记上一个是谁，$\sum b$，上一个 b，复杂度全错。
但是 b 真的需要记下来吗，发现 b 是多少转移都一样，只影响 $\sum b$ 的大小。

首先 $=m$ 没有意义，加到最后一个即可。
如果枚举排列 p，b 的构造是显然的贪心。
考虑直接状压，需要记上一个是谁，$\sum b$，上一个 b，复杂度全错。
但是 b 真的需要记下来吗，发现 b 是多少转移都一样，只影响 $\sum b$ 的大小。
转移时不妨直接枚举 $b_{p_{i+1}} - b_{p_i}$，把这部分贡献全加上。
$\mathrm{O}(2^n n^2 m)$

例题选讲

洛谷 P4229 某位歌姬的故事

计数有多少 n 个数，值域 $[1, A]$ 的整数数列 a，同时满足：

$\max\limits_{j=l_i}^{r_i} a_j = m_i$

$n, A \leq 9 \times 10^8, Q \leq 500$

洛谷 P4229 某位歌姬的故事

先考虑 n 小的情况，发现每个数字都有一个上限，一个 m 限制的区间中无法达到 m 的那些数显然和它无关

例题选讲

洛谷 P4229 某位歌姬的故事

先考虑 n 小的情况，发现每个数字都有一个上限，一个 m 限制的区间中无法达到 m 的那些数显然和它无关
于是每个不同的 m 可以拆分成若干不同的问题，对于每个 m 他的限制形如：区间内至少有一个数等于 m。

先考虑 n 小的情况，发现每个数字都有一个上限，一个 m 限制的区间中无法达到 m 的那些数显然和它无关

于是每个不同的 m 可以拆分成若干不同的问题，对于每个 m 他的限制形如：区间内至少有一个数等于 m。

考虑 DP，设 $dp_{i,j}$ 为前 i 个数，上一个取到 m 的位置为 j 的方案数，转移是简单的。

先考虑 n 小的情况，发现每个数字都有一个上限，一个 m 限制的区间中无法达到 m 的那些数显然和它无关

于是每个不同的 m 可以拆分成若干不同的问题，对于每个 m 他的限制形如：区间内至少有一个数等于 m。

考虑 DP，设 $dp_{i,j}$ 为前 i 个数，上一个取到 m 的位置为 j 的方案数，转移是简单的。

n 大时考虑离散化，一个区间内限制都为 m 那么没达到的方案数：$(m-1)^{len}$，达到的方案数：$m^{len}-(m-1)^{len}$。

例题选讲

一道经典题

给定长为 n 的序列 A 和正整数 L，求有多少个排列 p 满足 $\sum_{i=1}^{n-1}|A_{p_i}-A_{p_{i+1}}|\leq L$。

绝对值并不好处理，考虑先对 A 从小到大排序，将排列转化成插空档（注意到两端和中间不太一样），这样绝对值就没有了。

绝对值并不好处理，考虑先对 A 从小到大排序，将排列转化成插空档（注意到两端和中间不太一样），这样绝对值就没有了。但是这样值域在负数区间可能很大，考虑拆贡献，把每个 $A_i - A_{i-1}$ 的贡献分别计算，这样算出来的大小就是递增的了。

绝对值并不好处理，考虑先对 A 从小到大排序，将排列转化成插空档（注意到两端和中间不太一样），这样绝对值就没有了。但是这样值域在负数区间可能很大，考虑拆贡献，把每个 $A_i - A_{i-1}$ 的贡献分别计算，这样算出来的大小就是递增的了。设 $f_{i,j,k,l}$ 为处理到 i 有 j 段数，大小为 k，端点确定了 l 个的方案数，转移就是考虑放端点、放空隙、合并两段、新增一段、放在一段的一端，具体可以看代码。

例题选讲

CF1188D Make Equal

给出 n 个数字 $a_1, a_2, \ldots, a_n$ ，每次操作可以给其中一个数加上 2 的非负整数次幂。求最少的操作次数，使得这 n 个数相等。
$n \leq 10^5$，值域 10^{17}

例题选讲

CF1188D Make Equal

排序后相当于找一个 x 使 $\sum_{i=1}^{n} \operatorname{popcount}(x + a_n - a_i)$ 最小。

例题选讲

CF1188D Make Equal

排序后相当于找一个 x 使 $\sum_{i=1}^{n} \operatorname{popcount}(x + a_n - a_i)$ 最小。
考虑数位 DP，但是进位信息很难维护，但是注意到按照后 k 位排序后进位的是一个后缀，于是只用记后缀的大小就行。

谢谢大家